

System and Method for Selecting Data Providers

The present invention relates to the identification, prior to or during the transfer of data, of suitable communications channels for transferring data. Embodiments of the present invention may be applicable in situations where a user wishes to receive data such as video data by way of Video Streaming for example, or prior to or during the downloading of multimedia content over any of a variety of means such as xDSL, Wireless LAN, mobile etc.

10 Background

Of the many uses of networks such as the internet, one category of use that has been gaining significantly in popularity recently has been the use of networks for the exchange of data such as video or audio data or other media content. Increasingly, types of multimedia content (e.g. Video, Audio) are available in large quantities on the Internet. This Multimedia content can be streamed over a variety of IP Networks, either broadband or narrowband. More generally, data distribution may be achieved by way of data streaming or more general forms of downloading. Such distribution may be carried out in a peer-to-peer context, or from commercial multimedia-providing servers, and may be carried out using a variety of means such as xDSL, Wireless LAN, cable, mobile (GPRS or 3G) etc. It should be noted that xDSL covers many different variations of DSL ("Digital Subscriber Line"), such as ADSL ("Asymmetric"), HDSL ("High bit-rate"), and RADSL ("Rate-Adaptive"). Digital Subscriber Line technology is a well-known technology for bringing high-bandwidth information to homes and small businesses over ordinary telephone lines.

25

Unfortunately, receiving data from the Internet can sometimes be problematic, due to factors such as Packet Loss, Packet Delay, Jitter, a Server being down, and other factors, which severely affect time-critical applications such as Video Streaming. Often, a user tries to connect to a Server and finds either that this Server is down or severely busy, thus refusing video content. If the user does manage to connect to a server and the video content is finally being streamed, the quality may be very bad due to factors such as packet delay, referred to above. All these factors severely affect an end-user's experience, particularly with regard to receiving streamed audio

30

or video data over the Internet, and have thus delayed the take-up of these kinds of applications.

For an individual user who wishes to download large files or other items, or perhaps
5 to receive streamed data relating to a particular file or files, there may be a large
number of different data providers, such as commercial servers or peers, who may
be able to provide the required data. On account of the variety of means by which
the user may be connected to the network, the variety of types of data providers
from which data may be sourced, and other factors, the individual user may
10 encounter the problem of not knowing which data provider, from a "pool" of
available data providers such as Video Servers, will be capable of serving video
content or other such data over a reliable connection. Low reliability may be caused
by packet loss, jitter, packet delay, and other factors, all of which lower the chances
of achieving good quality downloading or data streaming.

15

In order to give an idea of how important it is to solve that problem, some "well-known" applications are mentioned below:

1) Peer-To-Peer applications like "Napster" (a music file-sharing web-site that has now been shut down) and "Kazaa" (www.kazaa.com).

20

2) Video Streaming Content providers, where a number of video servers are set-up somewhere on the network.

With reference to video servers, to which the present invention is of particular applicability, the usual approach to this problem has been to pick a default video
25 server without any prior knowledge of which Server will provide the most reliable
connection, which is dependent on where the end-user is located on the network, however some techniques already exist for choosing a server which is likely to be
faster or better in some way.

30 Prior-art:

Summaries of some related papers are given below:

"Measurement study of peer-to-peer file sharing systems": Authors: Saroiu S, Gummadi PK, Gribble SD (Dept. of Comput. Sci. & Eng., Washington Univ., Seattle, WA, USA),

Multimedia Computing and Networking 2002, 23-24 Jan. 2002, SPIE-Int. Soc. Opt. Eng, Proceedings of the SPIE - The International Society for Optical Engineering, vol.4673 pp 156-70

Summary: The popularity of peer-to-peer multimedia file sharing applications such as Gnutella and Napster has created a flurry of recent research activity into peer-to-peer architectures. This paper includes a detailed measurement study of the two popular peer-to-peer file sharing systems, namely Napster and Gnutella, and in particular seeks to characterise the population of end-user hosts that participate in these two systems. This characterisation includes the bottleneck bandwidths between these hosts and the Internet at large, IP-level latencies to send packets to these hosts, how often hosts connect and disconnect from the system, how many files hosts share and download, the degree of co-operation between the hosts, and several correlations between these characteristics.

"Handling multimedia objects in peer-to-peer networks": Authors: Kalogeraki V (HP Labs., Palo Alto, CA, USA), Delis A, Gunopulos D, Proceedings CCGRID 2002. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, 21-24 May 2002, IEEE Comput. Soc pp 438-9

Summary: This paper attempts to explain how the furnishing of video services on a peer-to-peer (P2P) network can not only offer viable alternatives to the mostly proprietary architectures used today for the delivery of video services, but it can also be done in a reliable and scalable manner. Building upon the approach of ad-hoc P2P networks of resources, a new architecture that can support the storage and retrieval of movies and/or video clips is proposed. The proposed configuration exploits the availability of high-performance links to networks, the usage of exclusive and partial indexing in peers, making nodes "aware" of the content in their own vicinity, replication of objects and caching of popular items, as well as full connectivity among servers whenever feasible. The architecture claims to use efficient indexing mechanisms for the retrieval of the multimedia objects, guarantees continuous operation in light of server failures and allows for the transparent population of new servers as well as the evolution of the furnished services and/or network resources with minimal disruption to the users. One key provision to realise such a P2P infrastructure is that the core peers (or servers) are expected to be linked via a low-latency and high-bandwidth network that is capable of effectively handling and delivering voluminous multimedia data. It is also anticipated that end-users should have sufficient connections to object servers.

"Peer-to-peer streaming media delivery": Author: Stolarz D, Proceedings First International Conference on Peer-to-Peer Computing, 27-29 Aug. 2001, IEEE Comput. Soc pp 48-52

- 5 Summary: Whatever definitions have been put upon it, peer-to-peer is said to be an effective rallying cry for a new way of doing things. Streaming media delivery is particularly susceptible to a peer-to-peer architectural approach. Peer-to-peer systems have been shown to reduce the bandwidth cost and increase the scalability of on-demand and streaming content on the Internet. Similar techniques can be used
10 to create a "virtual multicast", an application-layer implementation of the efficient sub-net broadcast features of network-layer multicasting.

- "On peer-to-peer media streaming": Authors: Dongyan Xu, Hefeeda M, Hambrusch S, Bhargava B (Dept. of Comput. Sci., Purdue Univ., West Lafayette, IN, USA), Proceedings
15 22nd International Conference on Distributed Computing Systems, 2-5 July 2002, IEEE Comput. Soc pp 363-71

Summary:

- In this paper, a peer-to-peer media streaming system is studied with the following characteristics: (1) its streaming capacity grows dynamically; (2) peers do not exhibit
20 server-like behaviour; (3) peers are heterogeneous in their bandwidth contribution; and (4) each streaming session may involve multiple supplying peers. Based on these characteristics, two problems are investigated: (1) how to assign media data to multiple supplying peers in one streaming session and (2) how to quickly amplify the system's total streaming capacity. The solution proposed to the first problem is an
25 optimal media data assignment algorithm OTS_{p2p} , which results in minimum buffering delay in the consequent streaming session. The solution to the second problem is a distributed differentiated admission control protocol DAC_{p2p} . By differentiating between requesting peers with different outbound bandwidth, DAC_{p2p} is said to achieve fast system capacity amplification; benefits all requesting peers in admission
30 rate, waiting time, and buffering delay; and creates an incentive for peers to offer their truly available out-bound bandwidth.

- Referring now to background patent literature, a system for server-side optimisation of data delivery is disclosed in US 6,112,239 (Kenner et al). Similar systems are
35 disclosed in US 6,502,125 and US 2003/0145007 (both also Kenner et al). In these

systems, users are provided with software which must be executed from their own machines in order to perform the optimisation. Similarly, in US 6,477,522 (Young), a system for optimising the downloading of files from the internet is disclosed in which an applet intercepts the request for the file and determines the best server to provide the file. Again, it is necessary for software to be installed in, and executed from, the user's machine.

With reference to the field of digital electronic game-playing between users connected via the internet, US 6,304,902 (Black et al) discloses a method for ensuring that the quality of data communications links between several game-playing users and any necessary server or servers is adequate for such game-playing. Game-playing generally involves two-way exchange of information between multiple players and a common server, which may be selected from a plurality of possible servers. One server acts as a matchmaker, and selects a few servers from the plurality of possible servers, and from these selects one as the server for the requested game. It will be noted that on account of the nature of such game-playing systems, the aim is to allow several users to connect to the same server concurrently, so a server is chosen in such a way as to facilitate its use by several users who may attempt to connect to it concurrently or one shortly after another.

Summary of the Invention

Contrary to the above, embodiments of the present invention aim to identify the "fastest", "nearest" or otherwise most appropriate server for an individual user, or the fastest, best or most reliable connection, in order that a connection may then be established by that individual user to a server that is not only appropriate for the multimedia content download or other such data exchange required by that user, but that is least likely to be slowed down or overloaded due to it also providing data to a large number of other users. According to preferred embodiments of the present invention, this identification process may be carried out without the user needing to install or execute specific software on his or her own machine.

According to the present invention there is provided a system for selecting a preferred data provider from a plurality of data providers, the system comprising:

means for receiving a request for data from a client;

means for receiving client identification data from said client;

means for identifying a plurality of data providers capable of providing data to said client;

5 means for providing said client identification data to said data providers;

means for instructing said data providers to perform the steps of:

(i) sending a test signal to said client;

(ii) receiving a return signal from said client;

10 (iii) obtaining a measure of the elapsed time between the sending of the test signal and the receipt of the return signal;

(iv) making a signal indicative of the elapsed time available to the system; and

(v) making a signal indicative of their remaining capacity available to the system;

15 means for receiving elapsed time signals and remaining capacity signals from said data providers;

means for selecting a preferred data provider on the basis of said signals; and

20 means for providing information relating to the identity of said preferred data provider to said client.

According to the present invention, there is also provided a method for selecting a preferred data provider from a plurality of data providers, the method comprising the steps of:

25 receiving a request for data from a client;

receiving client identification data from said client;

identifying a plurality of data providers capable of providing data to said client;

providing said client identification data to said data providers;

30 instructing said data providers to perform the steps of:

(i) sending a test signal to said client;

(ii) receiving a return signal from said client;

- (iii) obtaining a measure of the elapsed time between the sending of the test signal and the receipt of the return signal;
- (iv) making a signal indicative of the elapsed time available to the system; and
- 5 (v) making a signal indicative of their remaining capacity available to the system;

receiving elapsed time signals and remaining capacity signals from said data providers;

selecting a preferred data provider on the basis of said signals; and

- 10 providing information relating to the identity of said preferred data provider to said client.

By using systems and methods according to embodiments of the present invention, it is possible to solve the problems outlined above by running a small testing process
15 which may be completely invisible to the end-user and which does not require the installation or execution of any specific software on the user's machine, when the user wishes to try to access video content or other data, thus determining from a number of available Video Servers or other data providers the most reliable, fastest, least congested or otherwise most appropriate one for the particular user. In this
20 way, the user may receive the best possible data download or data stream by connecting to the most "reliable" one, thus enhancing the user's experience in relation to multimedia and other such applications.

According to preferred embodiments, the system may be arranged to receive
25 requests for specific items, such as specific video files, from the user. It may then carry out a search in order to identify data providers capable of providing the specific items requested, possibly of a group of data providers who may be pre-selected, or who may be subscribers to the system, or possibly of the whole internet or other such network. Following such a search, the system may perform the selection
30 process in order to make a selection just from these potential data providers. Alternatively the user need not be asked to specify a particular item in order for a "preferred" data provider to be selected. In such cases, however, the user may find him- or herself limited to a "catalogue" of items that the preferred data provider,

once selected, can provide, which may or may not include a specific item that the user wishes to request.

In systems according to preferred embodiments of the invention, the means for
5 instructing said data providers is a means remote from the client, such as a
centralised server. The term "remote" need not imply that the instructing means and
the client are geographically distant, but merely that the respective functions of the
client and the instructing means are not performed by a common processing means
on a client machine. It will be clear that irrespective of the respective physical
10 locations of the client and the server, something other than the client performs the
"instructing", thus there is no need for software to be downloaded to a client
machine.

Systems according to embodiments of the invention may be arranged to select a
15 preferred data provider only from data providers having a remaining capacity above a
predetermined threshold, effectively disqualifying any data providers whose
remaining capacity is below that predetermined threshold. Alternatively, the final
selection may be made in order to obtain the best balance between the respective
factors represented by the two types of signals (i.e. elapsed time and remaining
20 capacity) without the decision being constrained by specific thresholds.

According to preferred embodiments, the signal indicative of remaining capacity that
data providers may be instructed to provide may be a signal indicative of their
remaining bandwidth.

25

In systems according to preferred embodiments, in order to provide the information
relating to the identity of the preferred data provider, the system may be arranged to
provide the information on a web site, which may be updated whenever the
selection process has been carried out and a new "preferred" data provider
30 established. The information may be provided in the form of the Uniform Resource
Locator (URL) of the preferred data provider. The information may be provided in
other ways, however, such as the sending of an e-mail or other such message
containing the necessary information to the user.

Systems according to embodiments of the invention may select more than one "preferred" data provider. In this case, they may provide information relating to a plurality of preferred data providers to the user, possibly in the form of a list
5 indicating a ranking, based on the order in which they performed in the respective tests (i.e. best, second-best etc.), or alternatively, any that performed above predetermined quality thresholds may be identified to the user.

Reference will be made to the following technologies in the description of preferred
10 embodiments of the invention: RMI, JAVA, Servlets, HTML. Information about these technologies is publicly available, but a brief summary is given here for the purposes of avoiding the possibility of ambiguity in relation to the terminology, and the abbreviations and acronyms associated therewith.

15 RMI ("Remote Method Invocation") is a new Application Programming Interface (API) offered in Java Development Kit (JDK) 1.1 that allows for messaging between different Java Virtual Machines (JVMs), even if they are separated by a network.

"JAVA" is a programming language expressly designed for use in the distributed
20 environment of the Internet. It can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module or "applet" for use as part of a Web page. Applets make it possible for a Web page user to interact with the page.

25 A "Servlet" can be defined as a small program that runs on a server. The term usually refers to a Java applet that runs within a Web Server environment. This is analogous to a Java applet that runs within a Web Browser environment.

HTML ("Hypertext Markup Language") is the set of symbols or codes inserted in a
30 file intended for display on a World Wide Web browser page. The "Markup" tells the Web browser how to display a web page's words and images for the user.

Brief Description of the Drawings

Further features and advantages of the present invention will become apparent from the following description of embodiments thereof, presented by way of example only, and by reference to the accompanying drawings, wherein like reference
5 numerals refer to like parts, and wherein:

Figure 1 is an illustration showing the component parts of a network including a system according to an embodiment of the present invention.

Figure 2 shows the calculation of the remaining "up-link" capacity of a "Data
10 Provider".

Detailed Description

A network including a system according to an embodiment of the present invention is shown in Figure 1. This figure demonstrates the interactions between components of the network which may occur in the process of determining a preferred, or the
15 "best", Video Server. Part of this is typical architecture used for video streaming (Client, Web Server, Video Server) but it also includes the JAVA RMI running, which allows the testing to be done in order to determine which "Video Server" is the most suitable for Video Streaming for a specific end-user. The process will be described in detail below. The steps in the figure indicate the preferred order of events. The
20 components of the system can be summarised as follows:

The Client: The "Client" 10 generally refers to the "end-user's" personal computer (PC), running a web browser and "Video Player" or a similar plug-in or application. It will be noted that the client could however be a device such as a 3G ("Third-
25 Generation") mobile phone, using WAP (Wireless Application Protocol) or similar web browsing protocols to interact with the internet, for example.

The Centralised Server: The "Centralised Server" 20 generally refers to a computer terminal such as a PC, comprising a Web Server 22 running web server software.
30 The Centralised Server 20 may thus provide or present information to the user in the form of a web page, from which the user may make a choice of video clips, for example, the web page containing links to video streaming server sites. The Centralised Server may run "Servlet" (JAVA) software 24 or "ASP" (Microsoft)

software, responsible for creating dynamic web pages and communicating with RMI servers, and finally, running JAVA RMI client, in order to communicate with "RMI servers" installed in any of a plurality of "Video Streaming Servers".

- 5 The Centralised Server 20 may thus initially present a dynamic web-page to the user. Prior to the Centralised Server operating in any "search" mode in which it aims to find the "best" (i.e. the fastest, nearest or otherwise most appropriate) server or servers, it may contain links to one or more "default" servers. At any point, the user may choose, or send a request for, a piece of content - if this occurs before any
10 search has been completed, a default streaming server may then be chosen to deliver the content, or a link to one or more default servers may be provided.

- Video Streaming Servers: A plurality of "Multimedia Servers" or "Video Streaming Servers" 30 are shown (in this example, three such servers are shown, identified as
15 "C1", "C2" and "Cn"), each of which contains, or has access to, stored "Multimedia Content" 35 such as "Video Content", compressed or uncompressed, an "RMI Server" 36 which communicates with the "RMI Client" 26 in the Centralised Server 20, and suitable software capable of serving "Video Streaming Content" to end-users such as the Client 10. They are also provided with means for carrying out a
20 "Ping" test, as will be explained later, and for establishing a value, which may be an average value, for a "round-trip response-time" as will be described in more detail later, and providing this to the Centralised Server.

- All of these components may interact in order to deliver video streaming content to
25 end-users, as will be explained later.

- The process of determining the "best" (or at least a "preferred") Server is described below. It will be understood that while "best" is a subjective term, two factors which are of great importance in data transfer are speed and reliability. Any
30 improvement in relation to either of these factors can be regarded as an improvement in the overall quality of the download. There are two principal aspects to the process, which will be referred to as follows:

- (a) a "Latency" test; and
- (b) a "Remaining Bandwidth" test.

The tests can be carried out one after the other in either order, or
5 contemporaneously. They will be described in the following paragraphs.

a) Latency Test

Step 1 (shown in Fig.1): The "Client" or "User" 10 submits a request for connection to a "Multimedia Server", via a "Centralised" Server 20, which is responsible for the
10 co-ordination of the entire "search" process for multimedia servers. This Centralised Server 20 contains a "Servlet" 24, which is capable of retrieving the IP address of the "Client" or "User" machine 10 in order that it may propagate this IP address to a number of Multimedia Servers 30, using "JAVA RMI" technology for example. The user may make a request for specific data, or a specific item, such as a specific
15 video file for example, in which case the Centralised Server may carry out a search for Video Servers capable of providing that data, item or file before continuing with the process of determining the "best" Server from those that are found to be capable thereof. Alternatively, the user's request may be for data in general, in which case the Centralised Server may carry out the process of determining the "best" Server
20 from a set of servers, predetermined or otherwise, in which case the user may be allowed to select which item or items to receive from the preferred server once the identity of that preferred server has been established, once that server has provided a "library" of available items to the user, for example.

25 Step 2 (shown in Fig.1): As referred to above, the IP address retrieved from the "Client" or "User" machine 10 is propagated from the "RMI Client" 26 of the Centralised Server 20 to the "RMI Servers" 36 of the Multimedia Servers 30.

Step 3 (shown in Fig.1): Each "RMI Server" 36 located in a "Video Streaming
30 Server" 30 will retrieve that IP address, and each one will send a test signal by "PING-ing" the "User" machine 10, using that IP address. "Ping" refers to the application software "Packet INternet Gophers" which may be used to operate the process of sending Internet Control Message Protocol (ICMP) packets from a "Video

streaming Server" machine to a "Client" machine, and in this way, it is possible to measure the time it takes for a packet to travel from that "Video Streaming Server" machine 30 to a "Client" machine 10 and return to the Video Streaming Server. Generally, The more successful packets received back (if more than one is sent) and the less time (generally to be measured in milliseconds) it takes for a packet to travel from a specific "Video Streaming Server" machine to the Client machine and back, the better the video-streaming performance that end-user is likely to get if connected to that "Video Streaming Server". At this point, it should be mentioned that a packet may be considered "unsuccessful", if a "Request Timeout Response" message is received after the "Ping" process for some packets. In this case, the packet is considered "lost" and a default value (usually 1000ms), may be given, thus affecting the "average" value calculated at the end.

While it is particularly advantageous to utilise the "Ping" test as above, in particular because it requires no extra software to be installed on the user machine, alternatives do exist for measuring latency. These alternatives include known network tools such as "Traceroute" and "Ping" equivalents suitable for protocols other than ICMP, such as UDP ("User Datagram Protocol").

Step 4 (shown in Fig.1): After the "Pinging" process, in each machine, has been completed, an average value is calculated and this value is returned back to the "Centralised" Server, using RMI again. Thus, a "table" with "average" response value times, like the ones shown in Table 1 will be formed in the "RMI Client". From all these values, the smallest one (i.e. Server C1 in this case) may be chosen as the preferred, or most suitable "Video Streaming Server" (or "best" Server).

IP address(Video Streaming Server)	Average "response time" value (milliseconds)
C1: 132.146.107.61	57 ms
C2: 132.146.107.124	1000 ms
Cn: 132.146.107.16	540 ms

Table 1: This shows the average values retrieved from all "Video Streaming Servers" with RMI technology. The "RMI Client" may pick the smallest value (i.e. Server C1).

- 5 Step 5 (shown in Fig.1): The "Servlet" then retrieves from the "RMI Client" the IP address of the "Video Streaming Server" with the smallest "average response time". In the above example this is "C1" with IP address "132.146.107.61". It may update the web page containing the video links with that new IP address. In this way, the "Centralised" server will redirect the client to that "multimedia" Server, via JAVA
- 10 "Servlet" technology.

This is the end of the latency test. The entire process as set out above may be invisible to the end-user, may take only a few seconds to be completed and by selecting on the basis of this test alone a Server may be selected from which the

15 user may get video streaming content from a "preferred" "Video Streaming Server".

The above process can be repeated after specific periods of time set by the "Video Server" administrator for example, and each time the web page can be dynamically refreshed with a new preferred "Video Streaming Server" IP address.

20

b) Remaining Bandwidth Test

While the system as described above is capable of establishing a preferred server on the basis of the results of the "latency test" alone, systems according to embodiments of the invention are also capable of performing a further test, which

25 will be referred to as the "Remaining Bandwidth Test". This allows a server to be "disqualified" from being chosen as the preferred server if it is currently "congested", due to being used by a significant number of other users, or due to a high proportion of its bandwidth already being assigned to other tasks. Figure 2 illustrates the calculation of the remaining "up-link" capacity in a "Server", for a

30 specific period of time.

Step 3a (not shown in Fig.1): Preferably, but not necessarily, at the same time as Step 3 of the "Latency Test", the "RMI Servers" 36 of each of the Multimedia

Servers 30 may obtain from the "Video streaming software" a value U for the number of other users already connected to that Multimedia Server, and for each one, the "bit-rate" B of the clip requested. With reference to Figure 2, the bit-rate of each of the U existing users is shown, for the sake of simplicity, as being 220
 5 kilobits per second, although the bit-rates of existing users need not be the same. Such information could be programmatically retrieved, using "plug-in" libraries like the "Windows Media SDK" tools from "Microsoft", or similar tools from other companies (RealVideo, QuickTime, etc).

10 The formula below will give the total bandwidth consumed at the time of the request from the "RMI Server":

$$N_{\text{total}} = \sum B_i, \text{ where } i = 0, 1, 2, 3, \dots, U \quad (\text{F.1})$$

where:

N_{total} is the Total Bandwidth consumed by the requested "Video streams", at the
 15 time the request from the "RMI server" took place.

B_i is the Encoding "Bit-Rate" of the requested "video clip", for the i^{th} "User".

U is the Number of connected "Users" to the "Video Streaming Server"

At the same time, the "RMI Server" may establish the maximum available
 20 "upstream" bandwidth for a "Video Streaming Server" limited by the network connection. This could be either set by the "administrator" manually, when he installs the entire software, or it could be retrieved automatically, from a process running locally on the machine, which determines the maximum "up-link" connection bandwidth.

25

Thus: X_{max} is the maximum available "upstream" bandwidth.

Finally, the formula below will give us the "percentage" of available "up-link" bandwidth:

30

$$A = [(X_{\max} - N_{\text{total}}) / X_{\max}] * 100\% \quad (\text{F.2})$$

where **A** is the percentage of remaining "up-stream" bandwidth

- 5 In this way, we can set a "threshold", of 10-20% for example, such that if **A** is below the threshold, we can assume that this "Video Streaming Server" is almost congested, thus it will not be included in the final "best Video streaming Server" decision (Step 4).
- 10 Below an example is given to explain the formulas above:

Let us take, for example, a situation where there are two clips encoded in a "Video Streaming Server": The first file is called "videofile1" and has an encoding bit-rate of 220kbps. The second is called "videofile2" and has an encoding bit-rate of 140kbps.

- 15 Ten other "Users" are already connected to the "Video Streaming Server", seven of which are watching "videofile1" and three of which are watching "videofile2". The maximum available bandwidth is $X = 10\text{Mbps} = 10000\text{Kbps}$. We set the threshold 20%.

- 20 From formula (F.1), we have the following:

Number of users: $U=10$

7 users watching "videofile1": $B1=220, B2=220, B3=220, B4=220, B5=220, B6=220, B7=220$

3 users watching "videofile2": $B8=140, B9=140, B10=140$

- 25 Thus, the total bandwidth consumed is:

$$\begin{aligned} N &= B1+B2+B3+B4+B5+B6+B7+B8+B9+B10 \\ &= 220+220+220+220+220+220+220+140+140+140=1960 \end{aligned}$$

$$N_{\text{TOTAL}} = 1960\text{kbps}$$

As set out above, the maximum available bandwidth $X = 10000\text{kbps}$

From formula (F.2):

$$\begin{aligned} A &= [(X-N)/X] * 100\% \\ &= [(10000 - 1960)/10000] * 100\% = 80.4\% \end{aligned}$$

so A=80.4%

- 5 Conclusion: The remaining available "up-link" bandwidth is 80.4%, above the threshold of 20%. So this server is capable of accepting requests for more "video streams" and it's "average response time" from Step 3, will be included in the final "best Video Streaming Server" decision.
- 10 It should be noted that alternative way of calculating the "remaining capacity" exist, such as the following. A program could be running continuously on the Server or other such Data provider which is capable of measuring the packets (TCP/UDP) sent out over a period of time, thus measuring the "Average up-link capacity". Such programs are widely available and would give an estimate of the traffic to and from
- 15 the Data Provider. Such processes may be more complicated than that described above, but may be capable of providing a more accurate measurement of the instantaneous average remaining bandwidth, and may also measure not only any "multimedia packets", but also any other traffic (TCP acknowledgement messages, overhead packets, traffic from other network applications etc). Such methods would
- 20 in general run continually on the "Data Provider", while the one described in detail above need be initiated only when the "bandwidth measurement" is required.

Step 4a (not shown in Fig.1): Once the above value has been established, the percentage of available "up-link" bandwidth may be returned to the RMI Client 26 of the Centralised Server 20 by each Video Streaming Server 30. Once the "Pinging" process (Step 3 of the "Latency Test") has also been completed in relation to each machine, a table with average response time values and percentage values may be formed in the "RMI Client" (see Table 2 below). If the percentage of available "up-link" bandwidth of any Video Streaming Server is below a predetermined

30 "threshold", that Server may be disqualified irrespective of its response time value. From those that are not disqualified, the one with the smallest average response

time value (i.e. "C1" in this example) may be chosen as the most suitable (or "best") Video Streaming Server.

IP address(Video Streaming Server)	Average response time (milliseconds)	Available "up-link" bandwidth (%)
C1:132.146.107.61	57ms	80.4%
C2:132.146.107.124	100ms	94%
Cn:132.146.107.16	54ms	15% (rejected)

Table 2: This shows the "Average response time" values retrieved from all "Video Streaming Servers" with RMI technology, together with the percentage of available up-link bandwidth values. The "RMI Client" may pick the Server having the smallest response time value that is not disqualified on account of having a percentage value below the predetermined threshold of 20%. Note that the n^{th} server (Cn) seems "congested", so its "average response time" value, although very low, will be rejected from the final decision.

Step 5 (shown in Fig.1): The "Servlet" will retrieve from the "RMI Client", the IP address of the preferred Video Streaming Server, and it will update the web page containing the video links with that new IP address. In this way, the "Centralised" server may redirect the client to that "multimedia" Server, via JAVA "Servlet" technology.

This is the end of the test, by virtue of which the user will be able to receive video streaming content from the "best" Video Streaming Server.

The process can be repeated for a fixed period of time, set by the "Video Server" administrator, and each time, the web page can be dynamically refreshed with a new "Video Streaming Server IP address".

Study of special cases:

Below, we briefly review some specific situations which may disrupt the above processes:

- RMI server is "down": In this case, the RMI Client may not be able to establish communication with the RMI Server of a particular Video Streaming Server. It may thus assume that this "Server" is currently not working. Thus, this "Server" will not be taken into account in the decision of which "Video streaming Server" is the "best".
- 5
- User/Client is behind firewall: In this special case, there is a possibility that the Client's machine blocks all "Ping" packets, resulting in all servers receiving "request time out responses". In this case, the end-user may be offered a default "Video Streaming Server" and may be informed about this event (i.e. that he is behind a firewall and should deactivate the blocking of ICMP packets). Alternatively, other processes may be automatically initiated to tackle this case.
- 10
- "Video Streaming Server" is "down": In this case, the RMI Server may check if the video streaming software is working and may inform the "RMI Client" if or when that "Video Streaming Server" is ready to receive connections. Alternatively, in Step 4, this "Server" may be excluded from the process of establishing which "Video Streaming Server" is the "best".
- 15
- 20